04-OPTIONAL -Milestone Project 1 - Advanced Solution

August 28, 2018

1 Tic Tac Toe - Advanced Solution

This solution follows the same basic format as the Complete Walkthrough Solution, but takes advantage of some of the more advanced statements we have learned. Feel free to download the notebook to understand how it works!

In [1]:
```
# Specifically for the iPython Notebook environment for clearing output
from IPython.display import clear_output
import random

# Global variables
theBoard = [' '] * 10   # a list of empty spaces
available = [str(num) for num in range(0,10)] # a List Comprehension
players = [0,'X','O']   # note that players[1] == 'X' and players[-1] == 'O'
```

In [2]:
```
def display_board(a,b):
    print('Available   TIC-TAC-TOE\n'+
           '  moves\n\n  '+
          a[7]+'|'+a[8]+'|'+a[9]+'        '+b[7]+'|'+b[8]+'|'+b[9]+'\n  '+
          '-----        -----\n  '+
          a[4]+'|'+a[5]+'|'+a[6]+'        '+b[4]+'|'+b[5]+'|'+b[6]+'\n  '+
          '-----        -----\n  '+
          a[1]+'|'+a[2]+'|'+a[3]+'        '+b[1]+'|'+b[2]+'|'+b[3]+'\n')
display_board(available,theBoard)
```

```
Available   TIC-TAC-TOE
  moves

  7|8|9         | |
  -----        -----
  4|5|6         | |
  -----        -----
  1|2|3         | |
```

In [11]:
```
def display_board(a,b):
    print(f'Available   TIC-TAC-TOE\n  moves\n\n  {a[7]}|{a[8]}|{a[9]}        {b[7]}|
display_board(available,theBoard)
```

```
Available   TIC-TAC-TOE
  moves

  7|8|9         | |
  -----        -----
  4|5|6         | |
  -----        -----
  1|2|3         | |
```

```
In [3]: def place_marker(avail,board,marker,position):
            board[position] = marker
            avail[position] = ' '

In [4]: def win_check(board,mark):

            return ((board[7] ==  board[8] ==  board[9] == mark) or # across the top
            (board[4] ==  board[5] ==  board[6] == mark) or # across the middle
            (board[1] ==  board[2] ==  board[3] == mark) or # across the bottom
            (board[7] ==  board[4] ==  board[1] == mark) or # down the middle
            (board[8] ==  board[5] ==  board[2] == mark) or # down the middle
            (board[9] ==  board[6] ==  board[3] == mark) or # down the right side
            (board[7] ==  board[5] ==  board[3] == mark) or # diagonal
            (board[9] ==  board[5] ==  board[1] == mark)) # diagonal

In [5]: def random_player():
            return random.choice((-1, 1))

        def space_check(board,position):
            return board[position] == ' '

        def full_board_check(board):
            return ' ' not in board[1:]

In [6]: def player_choice(board,player):
            position = 0

            while position not in [1,2,3,4,5,6,7,8,9] or not space_check(board, position):
                try:
                    position = int(input('Player %s, choose your next position: (1-9) '%(player
                except:
                    print("I'm sorry, please try again.")

            return position

In [7]: def replay():

            return input('Do you want to play again? Enter Yes or No: ').lower().startswith('y
```

```
In [ ]: while True:
            clear_output()
            print('Welcome to Tic Tac Toe!')

            toggle = random_player()
            player = players[toggle]
            print('For this round, Player %s will go first!' %(player))

            game_on = True
            input('Hit Enter to continue')
            while game_on:
                display_board(available,theBoard)
                position = player_choice(theBoard,player)
                place_marker(available,theBoard,player,position)

                if win_check(theBoard, player):
                    display_board(available,theBoard)
                    print('Congratulations! Player '+player+' wins!')
                    game_on = False
                else:
                    if full_board_check(theBoard):
                        display_board(available,theBoard)
                        print('The game is a draw!')
                        break
                    else:
                        toggle *= -1
                        player = players[toggle]
                        clear_output()

            # reset the board and available moves list
            theBoard = [' '] * 10
            available = [str(num) for num in range(0,10)]

            if not replay():
                break
```

```
Welcome to Tic Tac Toe!
For this round, Player X will go first!
```